

**М. Д. ГОДЛЕВСЬКИЙ**, д-р техн. наук, проф., зав. каф. АСУ НТУ «ХП»,  
**О. О. АБАБІЛОВ**, студент, каф. АСУ НТУ «ХП»

## РОЗРОБКА БІБЛІОТЕКИ ПАРАЛЕЛЬНИХ ГЕНЕТИЧНИХ АЛГОРИТМІВ ORAPGA ДЛЯ РОБОТИ В MPI-СЕРЕДОВИЩІ

У статті розглядаються вимоги до бібліотеки паралельних генетичних алгоритмів. Описано розроблену бібліотеку orapga, що задовольняє цим вимогам. Наводиться діаграма варіантів використання та діаграма класів. Перелічено варіанти генетичних операторів, які надає бібліотека. Описано процес налаштування orapga за допомогою конфігураційного файлу.

В статье рассматриваются требования к библиотеке параллельных генетических алгоритмов. Описана разработанная библиотека orapga, удовлетворяющая этим требованиям. Приводится диаграмма вариантов использования и диаграмма классов. Перечислены варианты генетических операторов, которые предоставляет библиотека. Описывается настройка orapga с помощью конфигурационного файла.

The article discusses the requirements for a library of parallel genetic algorithms. The developed orapga library satisfying these requirements is described. Use case and classes diagrams are given. Variants of genetic operators provided by the library are enumerated. Orapga setup process using a configuration file is described.

**Вступ.** На сьогоднішній день генетичні алгоритми довели свою конкурентноздатність при вирішенні багатьох NP-складних задач і особливо в практичних додатках, де математичні моделі мають складну структуру [1]. Тому вони зручні та продуктивні для розв'язання задачі створення розкладу, яка залишається однією з актуальних проблем в НТУ «ХП». Ідея генетичних алгоритмів запозичена у живої природи і полягає в організації еволюційного процесу [2], кінцевою метою якого є отримання оптимального рішення в складній комбінаторній задачі. Нами була обґрунтована доцільність застосування паралельних генетичних алгоритмів для рішення задачі складання розкладу, а також запропонована математична модель. Описано структуру особини й три типи паралельних генетичних алгоритмів для рішення задачі складання розкладу, виділені параметри, які впливають на ефективність описаних алгоритмів [3].

Для розв'язання задачі складання розкладу є доцільним відділити логіку предметної області від реалізації паралельних генетичних алгоритмів. Це дасть змогу:

- повторно використовувати бібліотеку генетичних алгоритмів для інших задач;
- покращувати реалізацію логіки предметної області без необхідності повторної компіляції бібліотеки генетичних алгоритмів;
- краще розподілити код по модулях, що спростить розробку та налагодження.

**Вимоги до бібліотеки.** Бажана бібліотека повинна мати наступні властивості:

- надавати широкий вибір варіантів генетичних операторів;
- дозволяти легко розширювати набір генетичних операторів;
- дозволяти конфігурувати отримане застосування, вказуючи у окремому файлі, які саме варіанти генетичних операторів потрібно використати та з якими параметрами;
- для легкості застосування більшість параметрів повинні мати значення за замовчуванням;
- реалізовані генетичні алгоритми мають бути паралелізовані з використанням бібліотеки MPI [4];
- мова реалізації – С або С++;
- залежностей від сторонніх бібліотек має бути якомога менше.

Вимоги із точки зору кінцевого користувача наведено на діаграмі варіантів використання (див. рис. 1).

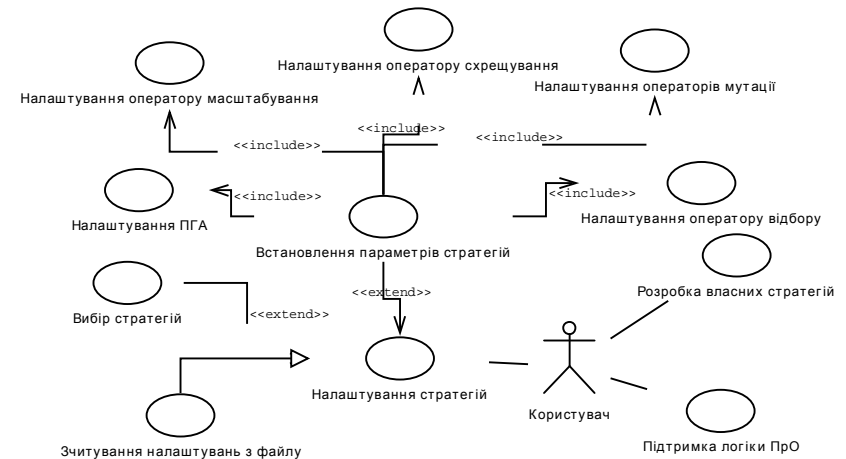


Рис. 1 – Діаграма варіантів використання

**Опис бібліотеки orapga.** Бібліотека orapga призначена для використання програмістами для будь-яких задач, в яких доцільно застосування генетичних алгоритмів. Програміст-користувач orapga має можливість налаштувати наявні стратегії (оператори масштабування, вибору, схрещування та ін.), в тому числі зчитати налаштування з файлу. Крім того, можлива розробка власних стратегій, в тому числі тих, що забезпечують підтримку логіки конкретної предметної області.

Бібліотеку orapga написано мовою С++. Створено складну ієрархію класів (див. рис. 2).

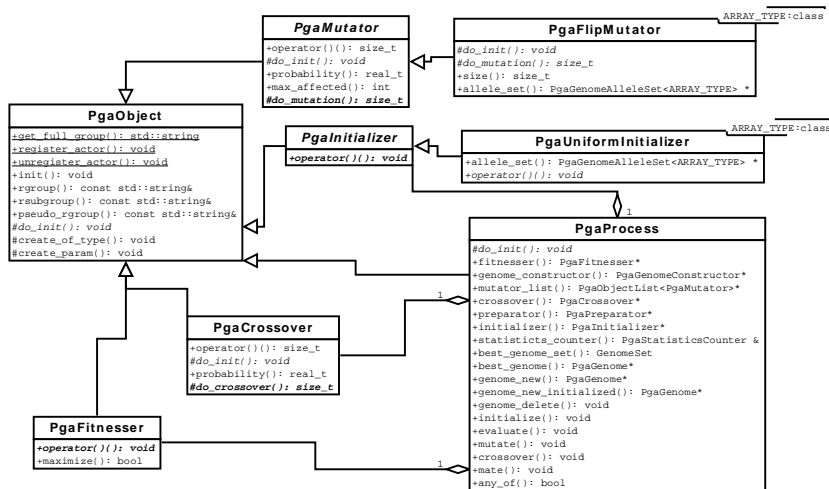


Рис. 2 – Фрагмент діаграми класів oarpga

Базовий клас для більшості класів – PgaObject. Цей клас забезпечує можливість динамічного створення об'єкту за іменем його класу та встановлення параметрів, що зберігаються в окремому об'єкті класу PgaParameterList. Останній клас зберігає іменовані параметри типів int, double, void\*, bool, char та string та дозволяє зберегти себе або зчитати із текстового файлу формату X Resources [5], а також передати себе іншому процесу через MPI.

Основні потомки класу PgaObject:

- PgaFitter – базовий клас для визначення придатності особини;
- PgaCrossover – базовий клас операторів схрещування;
- PgaMutator – базовий клас операторів мутацій;
- PgaInitializer – базовий клас для ініціалізації особини;
- PgaSelector – базовий клас операторів відбору;
- PgaScaler – базовий клас операторів масштабування;
- PgaPopulation – базовий клас популяції;
- PgaCondition – базовий клас довільної умови, якій може задовольняти популяція;
- PgaProcess – базовий клас для основного процесу генетичного алгоритму.

Ці класи мають потомків, які реалізують конкретні стратегії. Наприклад, у PgaMutator є нащадок PgaFlipMutator – мутація flip.

Перелічимо основні варіанти генетичних операторів, реалізованих у oarpga:

- оператор відбору: roulette wheel, rank, tournament, uniform;
- оператор схрещування: even odd, one point, two point, uniform;
- оператор мутації: flip, swap;
- оператор масштабування: no scaling, linear.

Бібліотека oarpga містить три реалізації генетичних алгоритмів:

- непаралельний (зручний для налагодження програми);
- алгоритм, що використовує острівну модель (Multiple demes);
- алгоритм із динамічними демами (Dynamic demes).

Користувач має змогу як використовувати наявні класи стратегій, так і розробляти свої.

Генетичний пошук налаштовується за допомоги файлу формату X Resources. Для цього класи-нащадки PgaObject мають дві характеристики:

- рольова група: mutator, crossover тощо;
- рольова підгрупа: two\_point, flip, swap тощо.

Щоб задати, який варіант генетичного оператора треба використовувати, потрібно вказати групу та підгрупу. Наприклад, для використання двох мутацій (спочатку flip, потім swap) потрібно вказати:

pga.mutator: flip,swap

Параметри генетичних операторів задаються наступним чином.

Для всіх об'єктів PgaFlipMutator встановити probability рівною 0.3:

pga.mutator.flip.probability: 0.3

Для всіх об'єктів PgaMutator встановити probability рівною 0.2:

pga.mutator\*probability: 0.2

Якщо параметр не задано явно, буде використано значення за замовченням.

**Висновки.** Створено бібліотеку паралельних генетичних алгоритмів, яка надає широкий вибір готових генетичних операторів, дозволяє легко додавати нові класи та має зручний та потужний механізм налаштування генетичного пошуку. Розроблена бібліотека була використана у застосуванні пошуку розкладу занять вузу.

Розвитком даної бібліотеки є розширення набору готових класів та використання у різноманітних предметних областях.

**Список літератури:** 1. Holland, John H. Adaptation in Natural and Artificial Systems. Ann Arbor, MI : University of Michigan Press, 1975. – 228 p. 2. Darwin, Charles. On the Origin of Species by Means of Natural Selection. John Murray, 1859. – 502 p. 3. Годлевський М. Д., Абабілов О. О. Розробка та налаштування паралельних генетичних алгоритмів для розв'язання задачі створення розкладу занять вузу на основі GRID-системи // Вісник Національного технічного університету «ХП». Збірник наукових праць. Тематичний випуск «Системний аналіз, управління та інформаційні технології». – X. : НТУ «ХП». – 2010. – №67. – С. 3-7. 4. Snir, Marc. MPI : The Complete Reference / Cambridge, Massachusetts : The MIT Press, 1996. – 350 p. 5. Р. Петерсон. Linux : руководство по операционной системе / Пер. с англ. – К. : Издательская группа BHV, 1998. – 480 с.

Надійшла до редколегії 16.06.2011